VERIACT: ACTION-GROUNDED VERIFICATION FOR EFFICIENT INTERACTIVE VISUAL REASONING

Anonymous Submission
Paper under double-blind review

ABSTRACT

Vision-language models can reason effectively about static visual inputs, yet interactive visual reasoning remains challenging because errors in perception or action can propagate across subsequent decisions. An agent may fail to notice the effect of an action, repeat ineffective actions, or continue planning from stale beliefs, wasting costly environment interactions. We introduce **VeriACT**, a task-agnostic interaction framework that makes verification an explicit and consequential part of the agent loop. VeriACT requires each action to state a falsifiable expectation and, before the next action, verifies the resulting environment transition against that expectation. It combines model-based verification with controller-grounded visual signals, including frame-change and repetition detection, multi-frame transition analysis, click-target confirmation, and terminal-state verification. Verification outcomes can trigger replanning and conditional memory updates, allowing new evidence to directly modify the agent's subsequent behavior. We evaluate VeriACT on ARC-AGI-3 against a free-form control while holding the vision-language model, observation interface, memory, and evidence tools fixed. In a 13-game evaluation, VeriACT improves mean RHAE from 1.72 to 6.36 and reduces environment actions by 33.3% on levels successfully completed by both conditions, at the cost of increased inference-time computation. These results suggest that explicit, action-grounded verification can trade additional inference-time computation for greater task progress and fewer environment interactions, even when the underlying reasoning model remains unchanged.

1 INTRODUCTION

Vision-language models (VLMs) have achieved strong performance on complex reasoning tasks, but their success on static benchmarks does not necessarily transfer to *interactive visual reasoning*. In environments such as ARC-AGI-3, agents must infer unfamiliar rules from visual observations, test hypotheses through interaction, maintain knowledge across multiple steps, and adapt their behavior as new evidence appears (ARC Prize Foundation, 2026). Unnecessary exploration, repeated mistakes, and stale assumptions are particularly costly because performance depends not only on task completion but also on interaction efficiency.

A recurring failure in our analysis of free-form trajectories is that agents often act without explicitly verifying whether the resulting observation matches their expectations. They may miss visual changes, repeat ineffective actions, or continue planning from stale beliefs. This is especially problematic because each action is both a decision and an experiment: its outcome provides evidence about the environment and the validity of the agent's current hypothesis. If that evidence is not verified and incorporated, an error can propagate across subsequent actions.

We therefore ask: *can explicit, action-grounded verification help a fixed VLM use its existing capabilities more effectively?* The key idea is to verify the outcome of each action before committing to the next decision. An agent should state what it expects to observe, compare that expectation with the resulting transition, and use discrepancies to revise its plan or memory. This closes the loop between action, observation, and reasoning.

We introduce **VeriACT**, a task-agnostic framework that makes action-grounded verification an explicit and consequential part of the agent loop:

$$\text{OBSERVE} \rightarrow \text{VERIFY+PLAN} \rightarrow [\text{UPDATE}] \rightarrow \text{ACT}.$$

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

At each turn, the agent records the current observation, compares the previous action's outcome with its expectation, and determines whether its plan remains justified. When necessary, UPDATE synchronizes its working memory. The controller additionally provides grounded signals, including frame-change and repetition detection, multi-frame transition information, click-target checks, and level-boundary evidence. Every action is paired with a falsifiable expectation, creating an explicit loop from prediction to environment transition to verification.

We evaluate VeriACT on a subset of ARC-AGI-3 benchmark using the same underlying VLM, observation rendering, external memory, and evidence tools across conditions. A free-form control measures the overall effect of the framework. We measure both task progress, interaction efficiency, and cost including actions required per completed level, RHAE, and tokens spent.

Our contributions are threefold:

- We introduce **VeriACT**, a task-agnostic framework for action-grounded verification, where each action is paired with an explicit expectation and its observed consequence is verified before the next action.
- We develop controller-grounded transition evidence and verification-triggered control mechanisms that make discrepancies consequential, enabling synchronization, replanning, and memory updates before subsequent actions.
- We provide a controlled comparison on ARC-AGI-3 against a free-form control while holding the underlying VLM and shared interaction substrate fixed, jointly analyzing task progress, environment-action efficiency, verification activity, and inference-time overhead.

2 Related Work

Interactive Visual Reasoning and Multimodal Agents. Language models are increasingly used as agents that reason, act, and use tools. ReAct interleaves reasoning with task-specific actions Yao et al. (2023b); Tree of Thoughts searches over alternative reasoning paths Yao et al. (2023a); Reflexion uses feedback-driven reflection across trials Shinn et al. (2023); and Self-Refine iteratively revises outputs using self-generated feedback Madaan et al. (2023). Toolformer enables API use Schick et al. (2023), while HuggingGPT coordinates specialized models through an LLM controller Shen et al. (2023).

Related work extends these ideas to multimodal and embodied settings. SayCan grounds language-model plans in robot-skill affordances Ahn et al. (2022), while Inner Monologue incorporates environment feedback into closed-loop planning Huang et al. (2023). PaLM-E, RT-1, RT-2, VIMA, Open X-Embodiment, and OpenVLA scale multimodal embodied learning across sensory inputs, language, and actions Driess et al. (2023); Brohan et al. (2023); Zitkovich et al. (2023); Jiang et al. (2023); Collaboration (2023); Kim et al. (2025), while Voyager combines interaction, feedback, and persistent skills for long-horizon embodied learning Wang et al. (2024a).

Multimodal reasoning follows a parallel trajectory. Chain-of-Thought elicits intermediate reasoning and self-consistency aggregates multiple sampled reasoning paths Wei et al. (2022); Wang et al. (2023). Multimodal-CoT and Visual Chain-of-Thought extend explicit reasoning to visual inputs Zhang et al. (2024); Chen et al. (2024); subsequent work studies longer multimodal reasoning chains Dong et al. (2025) and more explicitly grounded intermediate reasoning Wu et al. (2026); Zhou et al. (2026).

Agent benchmarks span embodied, web, tool-use, and desktop environments, including ALFWorld, WebShop, WebArena, Mind2Web, VisualWebArena, AgentBench, GAIA, and OSWorld Shridhar et al. (2021); Yao et al. (2022); Zhou et al. (2024); Deng et al. (2023); Koh et al. (2024); Liu et al. (2024); Mialon et al. (2024); Xie et al. (2024). Building on ARC's emphasis on skill acquisition and generalization Chollet (2019); Chollet et al. (2025), ARC-AGI-3 introduces interactive visual environments in which agents must infer goals and dynamics without explicit task instructions while acting efficiently ARC Prize Foundation (2026). VeriACT targets a failure specific to such settings: agents may continue acting without checking whether an action produced its expected outcome. We address this without changing the underlying VLM by making action verification part of the interaction loop.

Verification, Reflection, and Process Supervision. Verification and self-evaluation are widely used to improve reasoning reliability. Process-supervision methods such as Let's Verify Step by Step and Math-Shepherd evaluate intermediate reasoning rather than only final answers Lightman et al. (2023); Wang et al. (2024b). Self-Refine uses critique to iteratively revise outputs Madaan et al. (2023), Reflexion uses verbal reflection to improve subsequent decisions Shinn et al. (2023), and CriticGPT trains models to identify errors in generated outputs McAleese et al. (2024). Constitutional AI likewise incorporates self-critique and revision during alignment Bai et al. (2022).

Multimodal work increasingly grounds or verifies intermediate reasoning. Visual Chain-of-Thought combines visual grounding, reasoning, and rationale confirmation, while Grounded Chain-of-Thought grounds intermediate steps in visual evidence Chen et al. (2024); Wu et al. (2026). RMultiplex200K develops multimodal process supervision and step-level verification Chen & Song (2025), and Guided Verifier provides intermediate verification signals during multimodal reasoning rollouts Sun et al. (2026). These methods primarily evaluate or ground reasoning steps and generated outputs rather than the observed consequence of an executed environment action.

VeriACT instead verifies an explicit prediction about an environment action against the observation produced by that action. Controller-grounded signals additionally capture frame changes, repeated actions, spatial targets, and level transitions. Because verification occurs before the next environment action, its outcome can force replanning or memory synchronization, closing the loop between *prediction, action, observation, verification, and subsequent planning*.

Agent Harnesses for ARC-AGI-3. Recent ARC-AGI-3 systems show that harness design can substantially shape interactive reasoning. VISTA provides a task-agnostic visual harness with rendered observations, visual evidence retrieval, persistent notes, and free-form reasoning and action Han et al. (2026). Its prompt encourages predicting action consequences and reporting subsequent changes, but does not enforce these checks as separate controller stages. AVO combines persistent memory and tools with supervision that detects stagnation and redirects strategy NVIDIA (2026). However, these evaluations do not isolate verification from other harness components. VeriACT uses a VISTA-shaped free-form loop as its control and introduces controller-enforced, action-grounded verification whose outcomes can trigger replanning and memory updates before the next environment action.

3 METHOD

We introduce **VeriACT**, a controller-enforced interaction framework that makes action verification explicit in visual reasoning. VeriACT treats each action as a hypothesis about how the environment should change: the agent predicts its observable consequence before execution, verifies the prediction against the resulting transition, and incorporates the evidence before acting again. This yields a closed-loop interaction process:

$$\text{Observe} \rightarrow \text{Predict} \rightarrow \text{Act} \rightarrow \text{Verify} \rightarrow [\text{Update}] \rightarrow \text{Act}.$$

VeriACT is task-agnostic and leaves the underlying VLM and tool set unchanged. Instead, the controller enforces this interaction structure and exposes trajectory-grounded evidence to support verification.

3.1 INTERACTION SETTING AND SHARED HARNESS SUBSTRATE

We consider the interactive visual setting of ARC-AGI-3 ARC Prize Foundation (2026). At turn t, the agent receives the rendered state I_t, available actions $\mathcal{A}_t$, environment status, and level progress, but no game rules, semantic labels, explicit objective, or textual board representation. It must therefore infer task dynamics and objectives through visual observation and interaction.

VeriACT and the free-form control share the same VISTA-shaped substrate (Han et al., 2026), including persistent conversation, external memories, archived visual evidence, and retrieval tools for prior frames, pixels, notes, and trajectory history. We call the control *VISTA-shaped* because we reimplement the relevant interface and tool semantics in a standalone harness. These capabilities are identical across conditions; VeriACT differs only in how the controller structures and validates successive interactions.

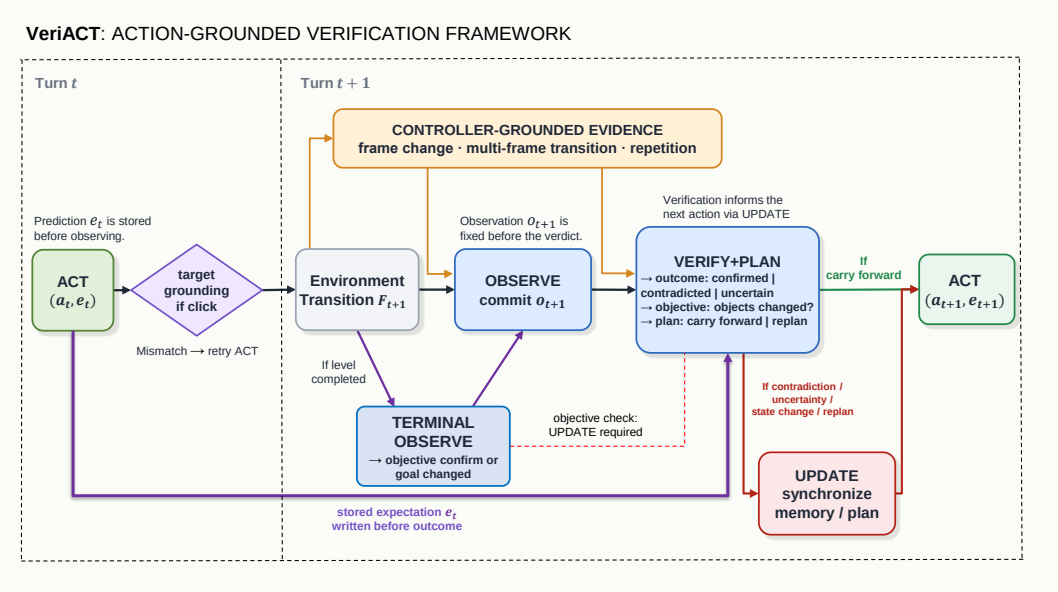


Figure 1: Overview of VeriACT's action-grounded verification cycle. Each action a_t is paired with an expected observable consequence e_t before execution. The resulting transition provides a committed observation and controller-grounded evidence, which VERIFY+PLAN uses to assess the action outcome, tracked state, and current plan. When the evidence requires revision, UPDATE synchronizes memory and plan before the next action. Coordinate-based actions additionally undergo target grounding, while level-completing transitions trigger terminal objective verification.

3.2 ACTION-GROUNDED VERIFICATION

At turn t, VeriACT requires the agent to select an action a_t together with an expected observable consequence e_t: (a_t, e_t). The expectation is committed before execution, forming a prospective claim about the transition that a_t should produce. For coordinate-based actions, the controller first checks whether the proposed location matches the intended target using a local visual crop, preventing coordinate-to-target inconsistencies from reaching the environment.

The validated action produces a sequence of frames

$$\mathcal{F}_t = \left(F_t^{(1)}, \ldots, F_t^{(K_t)}\right), \qquad I_{t+1} = F_t^{(K_t)},$$

where I_{t+1} is the settled state on the following turn. VeriACT retains the transition and derives controller-grounded evidence, including visual change, transient multi-frame effects, and recent action repetitions. These signals provide direct evidence about what occurred, reducing the need for the VLM to reconstruct the transition from its multimodal history.

On the following turn, the model first commits an observation $o_{t+1} = \textsc{Observe}(I_{t+1})$, before evaluating the preceding action. It may retrieve relevant memories and visual evidence, but cannot modify memory or act during observation. VERIFY+PLAN then compares the committed observation and transition evidence with the expectation e_t and returns $v_{t+1} \in \{\texttt{confirmed}, \texttt{contradicted}, \texttt{uncertain}\}$.

This produces the action-grounded chain $e_t \to a_t \to \mathcal{F}_t \to o_{t+1} \to v_{t+1}$. The controller enforces that the prediction is evaluated only after the corresponding action has produced its transition, while the semantic correctness of the model's verdict remains model-dependent.

In addition, beside describing the next received frame, if an action completes a level, a separate TERMINAL OBSERVE examines the completion transition and asks the model to reconfirm the inferred objective and whether its goal hypothesis has changed. The resulting objective information is passed to VERIFY+PLAN before the next plan is evaluated.

Verification then extends beyond the immediate action outcome to determine whether the agent's broader state and plan remain valid. The model compares o_{t+1} with the tracked working state,

Algorithm 1 VeriACT interaction loop

while run is active **do**
  if the previous action completed a level **then**
    TERMINALOBSERVE(completion transition)
  $o_t \leftarrow$ OBSERVE(I_t)
  $z_t \leftarrow$ VERIFYPLAN(o_t, e_{t-1}, g_t, M) ▷ g_t: controller-grounded evidence
  if NEEDUPDATE(z_t, event flags) **then**
    $M \leftarrow$ UPDATE(M, z_t)
  $(a_t, e_t) \leftarrow$ ACT(I_t, M)
  if a_t is coordinate-based **then**
    GROUNDTARGET(a_t)
  $\mathcal{F}_t \leftarrow$ ENVIRONMENTSTEP(a_t)
  $g_{t+1} \leftarrow$ COMPUTEEVIDENCE$(I_t, \mathcal{F}_t)$
  $I_{t+1} \leftarrow$ SETTLEDFRAME$(\mathcal{F}_t)$ “‘
end while

identifies relevant state changes and/or goal changes, and decides whether to carry the current plan forward or replan. Thus, an action may be confirmed while still requiring an update if its resulting evidence changes the agent's understanding of the environment.

3.3 VERIFICATION-TRIGGERED CONTROL

These verification results are not merely recorded: they determine which control transitions are permitted before the next action. When the outcome or associated evidence indicates that the persistent state or plan may be stale, the controller requires synchronization before execution can continue:

$$\text{NEEDUPDATE}_{t+1} = (v_{t+1} \neq \texttt{confirmed}) \vee \texttt{level_boundary} \vee (\texttt{mode} = \texttt{replan}) \vee \texttt{objects_changed} \vee \texttt{goal_changed}.$$

When triggered, UPDATE must synchronize external memories which store game and current level understanding before ACT becomes reachable. Replanning additionally requires an actionable replacement plan that accounts for previously ineffective attempts. Hence, the verification result has a concrete downstream consequence: transition evidence $\rightarrow$ verification $\rightarrow$ [update/replan] $\rightarrow$ next action. This distinguishes VeriACT from reflection performed after an action but outside the interaction protocol: the controller prevents the agent from executing another environment action until required consequences of the verification have been incorporated.

3.4 END-TO-END INTERACTION

The complete VeriACT cycle spans two consecutive turns. At turn t, the agent commits (a_t, e_t) and executes a_t after any required target grounding. The controller records the resulting transition and derives grounded evidence. At turn $t+1$, the agent commits the new observation, verifies the previous expectation, reassesses its tracked state and plan, and synchronizes memory when required. Only then can the next action be executed.

The complete dependency is

$$(a_t, e_t) \rightarrow \mathcal{F}_t \rightarrow \{g_{t+1}, o_{t+1}\} \rightarrow \text{VERIFY+PLAN} \rightarrow [\text{UPDATE}] \rightarrow (a_{t+1}, e_{t+1}),$$

where g_{t+1} denotes controller-grounded evidence derived from the realized transition. Algorithm 1 summarizes the resulting controller flow.

4 EXPERIMENTS

We evaluate whether making verification an explicit part of the interaction loop improves both task progress and environment-action efficiency while keeping the underlying vision–language model fixed. We focus on three questions: (1) whether VeriACT improves task progress and reduces environment interactions at matched successful progress, (2) whether its verification mechanisms

actively influence subsequent interaction, and (3) how much additional inference-time computation is introduced by this verification process.

4.1 Experimental Setup

Model and Inference Configuration. All experiments use `Qwen/Qwen3.8-27B` (Qwen Team, 2026), a dense 27B-parameter vision–language model with a 262,144-token context window. We use the public checkpoint without fine-tuning and enable its default thinking mode. Both the VISTA-shaped control and VeriACT use identical inference settings: temperature 1.0, top-p 0.95, a maximum of 32,768 output tokens, and reasoning effort `xhigh`. Inference is served with vLLM v0.19.1 through an OpenAI-compatible endpoint.

For game rendering, following VISTA (Han et al., 2026), each 64×64 environment frame is rendered as a 512×512 image with one-pixel grid lines drawn at 75

Tasks and Evaluation Protocol We evaluate on 13 ARC-AGI-3 games: `ar25`, `cd82`, `cn04`, `dc22`, `ft09`, `lp85`, `m0r0`, `re86`, `s5i5`, `sp80`, `sk48`, `su15`, and `tu93`. Interactive trajectories are substantially more expensive than single model calls, particularly for VeriACT, which may invoke multiple model stages per environment action. Both conditions use the same games, environment initialization, memory initialization, action/reset budgets, and stopping rules. Each run is capped at 300 interaction turns and terminates upon completion or another configured termination condition.

We compare full VeriACT against a matched VISTA-shaped free-form control while keeping the underlying VLM, observations, persistent memory, evidence-retrieval tools, and environment interface fixed. The control uses a single ungated interaction phase, whereas VeriACT enforces the OBSERVE→VERIFY+PLAN→[UPDATE]→ACT protocol described in Sec. 3. This matched setup isolates the effect of the interaction protocol while keeping the model and available interaction capabilities unchanged.

Metrics We report per-game RHAE as the primary task-performance metric, jointly capturing level completion and action efficiency by assigning zero credit to incomplete levels and higher scores to more efficiently completed levels. For environment-action efficiency, we report actions per completed level together with the corresponding human reference, allowing direct comparison of interaction cost at matched completed levels. For inference cost, we sum input and output tokens through the deepest level completed by both conditions, providing a matched comparison of model-side computation.

4.2 Overall Performance and Action Efficiency

Tab. 1 summarizes the comparison across the 13 games. VeriACT achieves a mean per-game RHAE of 6.36, compared with 1.72 for the VISTA-shaped control, and completes 14 levels compared with 9. The largest gains occur in environments where verification enables VeriACT to progress beyond levels not completed by the control, including `ft09`, `ar25`, and `tu93`. These results show that the benefit of verification is not limited to reducing redundant actions within already solved levels; it can also support continued task progress in challenging environments.

For a direct comparison of interaction efficiency, we consider the five levels successfully completed by both conditions: Level 1 of `ar25`, `cn04`, `lp85`, `re86`, and `s5i5`. VeriACT uses 104 environment actions compared with 156 for the control, reducing environment interaction by 33.3%. VeriACT requires fewer actions on four of the five matched levels and ties on the remaining level. Thus, the higher task progress is accompanied by lower environment-action cost when comparing the same successfully completed levels.

The reduction in environment interaction requires additional model-side computation. Across the same five jointly solved levels, VeriACT consumes 34.67M tokens compared with 15.12M for the control, corresponding to $2.29\times$ the inference cost, while using 104 rather than 156 environment actions. This illustrates the central trade-off of VeriACT: additional inference-time computation is used to explicitly verify actions, synchronize state, and revise plans, reducing the number of external interactions required to achieve the same completed progress.

	RHAE ↑		Actions / Level ↓			Tokens (Shared Solved) ↓	
Game	**Ctrl.**	**VeriACT**	**Human**	**Ctrl.**	**VeriACT**	**Ctrl.**	**VeriACT**
ar25	0.56	**8.33**	**32**/50	71	**32/29**	**9.56M**	13.18M
cd82	0.00	**4.76**	55	–	**46**	–	–
cn04	4.76	4.76	29	22	**21**	**1.20M**	5.67M
dc22	**2.80**	0.00	**59**	77	–	–	–
ft09	0.00	**47.62**	43/12/23/**28**	–	**6/7/14**/30	–	–
lp85	2.78	2.78	17	**7**	**7**	**0.37M**	1.11M
m0r0	**0.57**	0.00	**30**	87	–	–	–
re86	**7.92**	2.78	26/42	35/**38**	**25**	**2.47M**	6.92M
s5i5	2.52	**2.78**	20	21	**19**	**1.52M**	7.79M
sk48	0.00	0.00	**61**	–	–	–	–
sp80	**0.50**	0.00	**39**	120	–	–	–
su15	0.00	**2.22**	22	–	**20**	–	–
tu93	0.00	**6.67**	**19**/16	–	22/**15**	–	–
Mean	1.72	**6.36**	–	–	–	–	–

Table 1: Performance, environment-action efficiency, and inference cost on the 13-game evaluation. Under "Actions / Level," slash-separated entries correspond to successive completed levels. Each agent column reports only levels successfully completed by that condition. Human reports the ARC-AGI-3 reference action counts through the deepest level completed by either condition. If the condition doesn't complete a single level, **no data is shown**. Within each level, bold denotes the lowest action count among Human, Control, and VeriACT, with ties bolded jointly. "Tokens (Shared Solved)" reports cumulative token usage **only** through levels successfully completed by both conditions. The final row reports mean per-game RHAE.

4.3 Qualitative Analysis: Verification in an Evolving Interaction Model

Aggregate metrics show the effect of verification on task progress and interaction cost, but not how verification changes the trajectory itself. We therefore examine `ar25`, where the agent must align the gray reflection of a movable piece with a yellow target across a light-blue axis. Across Levels 1–2, VeriACT illustrates four complementary ways in which verification affects subsequent interaction: falsifying an action model, grounding ineffective actions in controller evidence, revising state and plans, and verifying objectives at level boundaries (Fig. 2).

Falsifying an inherited action model. At the start of Level 2, VeriACT initially assumes that directional actions control the black piece. At T33, it executes `ACTION4`, predicting that the black piece will move right. Instead, the black piece remains fixed while the central band and gray piece move. At T34, VERIFY+PLAN identifies the contradiction and replans around the hypothesis that the band is controlled (Fig. 2, top). The control encounters the same discrepancy but continues under the incorrect model through T75–T78. VeriACT therefore does not eliminate the initial error; rather, it makes the first contradictory transition consequential for subsequent action selection.

Terminating an ineffective probe. VeriACT next probes vertical movement with `ACTION2` at T34. The resulting frame is pixel-identical to the pre-action state, and the controller exposes this no-change evidence at T35. VERIFY+PLAN marks the prediction contradicted and switches strategy (Fig. 2, second row). The control instead issues five consecutive downward actions with unchanged outcomes before reaching the same conclusion. Here, the controller does not determine the strategy; it provides grounded evidence that the attempted action produced no observable transition.

Revising state and plan. Verification can also identify inconsistencies beyond the immediately preceding action. On Level 1, VeriACT's LEFT moves at T15–T16 succeed, but its state/plan check detects a mismatch between accumulated move counts and the measured state. It replaces hand-counted position tracking with direct measurement of the remaining gap, resets when this yields a shorter path, and follows the revised representation to completion (Fig. 2, third row). The control continues relying on accumulated move counts through repeated partial overlaps and resets before recognizing that its position estimate is inaccurate.

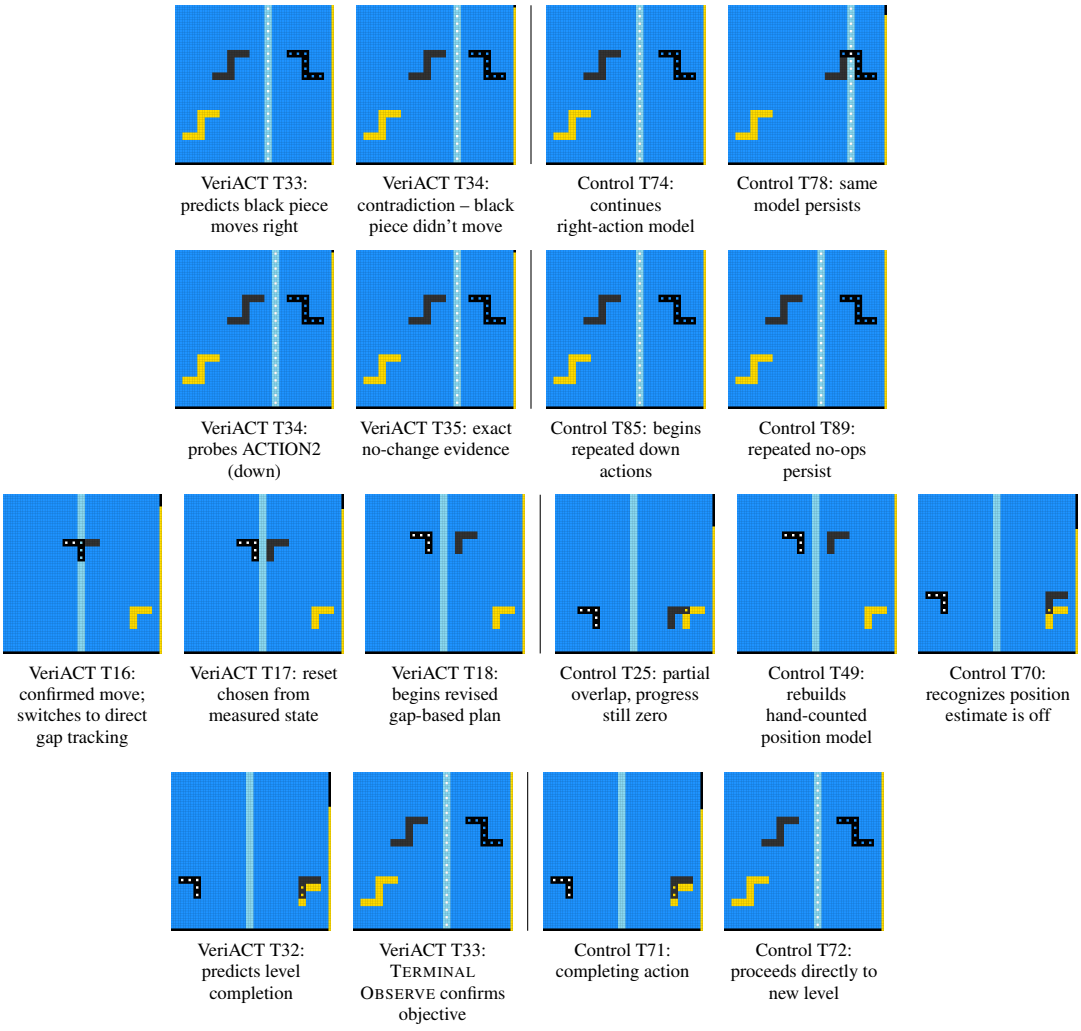


Figure 2: Qualitative trajectory from `ar25`. **Top:** VeriACT falsifies the inherited action model after one transition, whereas the free-form control continues under the same incorrect model. **Second:** a controller-computed no-change signal terminates an ineffective downward probe, while the control repeats the no-op. **Third:** despite locally confirmed moves, VeriACT detects inconsistency between its tracked and measured state, switches to direct gap tracking, and replans; the control continues relying on accumulated move counts. **Bottom:** at the Level 1→2 boundary, TERMINAL OBSERVE attributes completion to overlap between the gray reflection and yellow target before the next-level state is processed; the control transitions without an analogous boundary check.

Verifying objectives at level boundaries. At T32, VeriACT predicts that aligning the gray reflection with the yellow target will complete Level 1. Because the successful action immediately loads Level 2, TERMINAL OBSERVE preserves the completing transition and records the overlap as the objective before processing the new state (Fig. 2, bottom). The control reaches the same boundary at T71 but proceeds directly to the next layout without an analogous objective check. This allows VeriACT to preserve the task objective separately from the interaction model used to achieve it.

Together, these examples show how verification makes observed evidence consequential within the interaction loop. Action predictions expose incorrect transition models, controller signals ground observed effects, state checks revise broader representations, and boundary verification preserves task objectives across level transitions. The key distinction is therefore not whether the agent avoids initial mistakes, but whether evidence from those mistakes is incorporated before subsequent actions are selected.

5 Limitations and Conclusion

5.1 Limitations

Our evaluation focuses on ARC-AGI-3 as a challenging testbed for interactive visual reasoning, using a controlled setup with a shared VLM and interaction substrate. This design allows us to isolate the effect of the verification protocol, but the current study does not yet characterize performance across a broader range of models, environments, or interaction settings. In addition, verification introduces additional inference-time computation, reflecting an explicit trade-off between reasoning cost and environment interaction. A broader evaluation across model scales and environments would help characterize when this trade-off is most beneficial.

5.2 Conclusion

We introduced **VeriACT**, a task-agnostic interaction framework that makes action-grounded verification a required and consequential part of interactive visual reasoning. By pairing each action with an explicit expectation, verifying the resulting transition, and using the resulting evidence to synchronize state and control subsequent actions, VeriACT closes the loop between prediction, action, observation, and reasoning. In a controlled evaluation with the same VLM and shared interaction substrate, VeriACT achieves higher task progress and requires fewer environment actions on levels completed by both methods. These gains come with additional inference-time computation, reflecting a trade-off between internal reasoning and external interaction. Overall, the results suggest that making verification operational, rather than merely reflective, can improve interactive visual reasoning by allowing evidence from each action to directly influence subsequent decisions.

AI USE STATEMENT

Generative AI tools were used to assist with writing, editing, and proofreading during the preparation of this manuscript. We also use Generative AI for code generation. All AI-assisted content was reviewed and verified by the authors. The authors are responsible for the final content, methodology, experimental results, and conclusions presented in this paper.

ETHICS STATEMENT

This work does not involve human subjects, human-subject experiments, or the collection of personal or sensitive data. The experiments evaluate an interactive visual reasoning agent in benchmark environments designed for research purposes. The agent is restricted to the actions and interfaces provided by the evaluation environments, and the experiments do not involve deployment in real-world settings or autonomous interaction with external users or systems. We acknowledge that interactive visual agents may have broader applications, and that additional safety considerations would be necessary for deployment in real-world environments.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the VeriACT interaction framework, action-grounded verification procedure, controller-grounded visual signals, memory update mechanism, and experimental protocols in the paper and supplementary material. We also describe the baseline and staged ablation settings, environment interfaces, evaluation metrics, and model configurations used in our experiments. To facilitate reproduction, we will release the implementation, experiment configurations, and evaluation scripts upon publication, subject to the licensing and redistribution requirements of the benchmark environments and models used in this work.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Ryan Joshi, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Jornell Quiambao, Kanishka Rao, Sumedh Rettinghouse, Diego Reyes, Pierre Sermanet, Norman Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, Andy Zeng, Marina Zhang, Zhao Zhang, and Brianna Zitkovich. Do as i can, not as i say: Grounding language in robotic affordances. In *Robotics: Science and Systems*, 2022.

ARC Prize Foundation. Arc-agi-3: A new challenge for frontier agentic intelligence. *arXiv preprint arXiv:2603.24621*, 2026.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023.

Sijia Chen and Bin Song. Rmultiplex200k: Toward reliable multimodal process supervision for visual language models on telecommunications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

François Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. Arc-agi-2: A new challenge for frontier ai reasoning systems. *arXiv preprint arXiv:2505.11831*, 2025.

Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, 2023.

Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, 2023.

Qiushi Han, Keya Hu, Linlu Qiu, Cathy Wu, and Kaiming He. VISTA: A visual harness for reasoning in an interactive world. Project website, August 2026. URL `https://vista-research.github.io/`. Accessed September 23, 2026.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, 2023.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. In *International Conference on Machine Learning*, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, 2025.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. Llm critics help catch llm bugs, 2024. URL https://arxiv.org/abs/2407.00215.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: A benchmark for general ai assistants. In *International Conference on Learning Representations*, 2024.

NVIDIA. NVIDIA AVO reaches 100% on ARC-AGI-3, demonstrating a frontier-level general-purpose architecture for long-horizon autonomous agents. NVIDIA Technical Blog, August 2026. URL https://developer.nvidia.com/blog/nvidia-avo-reaches-100-on-arc-agi-3-demonstrating-a-frontier-level-general-purpose-architecture-for-long-horizon-autonomous-agents/. Accessed September 23, 2026.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 2023.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 2023.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Lingzhuang Sun, Ruitong Liu, Yuxia Zhu, Xiaohan Xu, Jingxuan Wei, Xiangxiang Zhang, Bihui Yu, and Wentao Zhang. Guided verifier: Collaborative multimodal reasoning via dynamic process supervision. *arXiv preprint arXiv:2602.04290*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024a.

Peiyi Wang, Lei Li, Zhihong Shao, R.X. Xu, Damai Dai, Yifei Li, Yuchuan Wu, Huazuo Wang, Xuebo Chen, Yidong Huang, et al. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2024b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Qiong Wu, Xiangcong Yang, Yiyi Zhou, Chenxin Fang, Baiyang Song, Xiaoshuai Sun, and Rongrong Ji. Grounded chain-of-thought for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, 2024.

Shunyu Yao, Howard Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. In *Transactions on Machine Learning Research*, 2024.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.

Yiyang Zhou, Haoqin Tu, Zijun Wang, Zeyu Wang, Niklas Muennighoff, Fan Nie, Chaorui Deng, Shen Yan, Haoqi Fan, Yejin Choi, James Zou, Cihang Xie, Huaxiu Yao, and Qinghao Ye. When visualizing is the first step to reasoning: Mira, a benchmark for visual chain-of-thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023.

APPENDIX